

CURRENT CLAIM STATUS AND CLAIM AMENDMENTS

Following is a set of all pending claims with current claim status and claim amendments to claims 8, 15-18, 20-21, and 25-26.

SUB B17

A^L

1. (original) A system for querying a database in response to data access requests issued by an application program, the system comprising:

a text file containing queries corresponding to said data access requests, wherein the queries are formatted in accordance with the syntax required by said database;

a database interface function;

a query lookup table, containing said queries, generated by operation of said database interface function on said text file;

wherein, in response to one of said data access requests, said database interface function submits, to the database, a query from said lookup table corresponding to said one of said data access requests, and retrieves the results of the query.

2. (original) The system of claim 1, wherein said database interface function also formats the results of the query.

3. (original) The system of claim 2, wherein said database interface function also returns the results of the query after the results have been formatted.

4. (original) The system of claim 1, wherein the queries in said query lookup table are located by name in response to one of said data access requests.

5. (original) The system of claim 4, wherein the query name indicates a type of data requested and a type of operation to be performed on the database.

6. (original) The system of claim 1, wherein each of the data access requests provided by the application comprises a query name and an argument array.

7. (original) The system of claim 1, wherein the queries in said text file are SQL strings.

8. (Currently Amended) A system for interfacing an application program with more than one type of database in response to data access requests issued by an application program, the system comprising:

A
a first text file containing queries corresponding to said data access requests, wherein the queries are formatted in accordance with said first database;

a database interface function;

a first query lookup table, containing said queries, generated by operation of said database interface function on said first text file;

a second database;

a second text file containing queries, formatted in accordance with said second database, corresponding to said data access requests; and

a second a query lookup table generated by operation of said database interface function on said text file;

wherein, in response to one of said data access requests issued to either said database, said database interface function submits, to the appropriate database, a query from said lookup table corresponding to said one of said data access requests, and retrieves the results of the query.

9. (original) The system of claim 8, wherein said database interface function also formats the results of the query.

10. (original) The system of claim 9, wherein said database interface function also returns the results of the query after the results have been formatted.

11. (original) The system of claim 8, wherein the queries in said query lookup table are located by name in response to one of said data access requests.

12. (original) The system of claim 8, wherein each of the data access requests provided by the application comprises a query name and an argument string.

13. (original) A method for querying a database in response to data access requests issued by an application program, the method comprising the steps of:

storing queries, corresponding to said data access requests, in a text file wherein the queries are formatted in accordance with the syntax required by said database;

reading said text file to generate a query lookup table containing said queries;

submitting, to the database, in response to one of said data access requests, one of said queries from said query lookup table corresponding to said one of said data access requests; and

retrieving the results of the query.

14. (original) The method of claim 13, including the additional step of formatting the results of the query.

15. (Currently Amended) The ~~system~~ method of claim 14, including the additional step of returning the results of the query to the application program after the results have been formatted.

16. (Currently Amended) The ~~system~~ method of claim 15, wherein the queries in said query lookup table are located by name in response to one of said data access requests.

17. (Currently Amended) The ~~system~~ method of claim 15, wherein each of the data access requests provided by the application comprises a query name and an argument array.

18. (Currently Amended) The ~~system~~ method of claim 13, wherein the queries in said text file are SQL strings.

19. (original) A method for querying a database in response to data access requests issued by an application program, the method comprising the steps of:

- A
- writing database queries to access the database;
 - storing said database queries in a text file;
 - reading said queries in said text file;
 - hashing said queries to generate a query lookup table containing said queries in hashed form;
 - receiving, from the application program, one of said data access requests including at least one parameter associated therewith;
 - locating, in the query lookup table, a selected one of the queries corresponding to said one of said data access requests received from the application program;
 - substituting said at least one parameter into corresponding positions in the selected one of the queries;
 - submitting, to the database, in response to one of said data access requests, said selected one of the queries; and
 - retrieving the results of the query.

20. (Currently Amended) The ~~system~~ method of claim 19, wherein the queries in said query lookup table are located by name in response to one of said data access requests.

21. (Currently Amended) The ~~system~~ method of claim 19, wherein each of the data access requests provided by the application comprises a query name and an argument string.

22. (original) A method for querying a database in response to data access requests issued by an application program, the method comprising the steps of:

creating a database interface function to handle said data access requests;

writing database queries to access the database;

executing said database interface function to perform the steps of:

storing the database queries as query strings in a text file;

reading the query strings in said text file;

hashing said query strings to generate a query lookup table containing said queries in hashed form;

receiving, from the application program, one of said data access requests including at least one parameter associated therewith;

locating, in the query lookup table, a selected one of the queries corresponding to said one of said data access requests received from the application program;

substituting said at least one parameter into corresponding positions in the selected one of the queries;

submitting, to the database, in response to one of said data access requests, said selected one of the queries; and

retrieving the results of the query.

23. (original) The method of claim 22, performed in an object-oriented programming environment wherein the database interface function comprises:

a parent class that handles application logic common to all queries;

and

a child class that formats the results of the query.

24. (original) The method of claim 23, wherein static data in said query lookup table comprises a class attribute available to all subclasses of said

parent class without requiring reloading of said query lookup for each of said data access requests.

25. (Currently Amended) The ~~system~~ method of claim 24, wherein the queries in said query lookup table are located by name in response to one of said data access requests.

26. (Currently Amended) The ~~system~~ method of claim 22, wherein each of the data access requests provided by the application comprises a query name and an argument string.
